2012 IEEE International Conference on Robotics and Automation
RiverCentre, Saint Paul, Minnesota, USA
May 14-18, 2012

Target-directed Navigation using Wireless Sensor Networks and Implicit Surface Interpolation

Nikhil Deshpande*, Edward Grant*, Thomas C. Henderson†

*Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC, USA. Email: {nadeshpa, egrant}@ncsu.edu

†School of Computing, University of Utah, Salt Lake City, UT, USA. Email: tch@cs.utah.edu

***Abstract*—This paper extends the novel research for event localization and target-directed navigation using a deployed wireless sensor network (WSN) [4]. The goal is to have an autonomous mobile robot (AMR) navigate to a target-location by: (i) producing an artificial magnitude distribution within the WSN-covered region, and (ii) having the AMR use the pseudo-gradient from the *interpolated* distribution in its neighborhood, as it moves towards the target location. *Implicit* surfaces are used to interpolate the artificial distribution. This scheme only uses the topology of the WSN and received signal strength (RSS) to estimate an efficient navigation path for the AMR. Here, the AMR does not require global coordinates for the region, as it relies on local, neighborhood information alone to navigate. The performance of the scheme is analyzed with hardware experiments and in simulation, using a variety of node-densities and with increasing levels of noise to ensure robustness.**

***Index Terms*—Target-directed navigation, pseudo-gradient, spline-interpolated distribution, received signal strength**

I. Introduction

Autonomous mobile robots (AMRs) operating in unknown and unstructured environments are confronted with fundamental challenges: (i) the localization of targets within a region; and (ii) navigating towards identified target locations in an efficient manner. The target locations can be emergent events, such as fires, chemical leaks, accidents, natural disasters, search-and-rescue, etc., or they can be pre-planned, such as area exploration, agricultural operations, robotic area cleaning, etc. In previous work [4], a novel scheme was presented for target-directed navigation within a distributed WSN. In brief, that research:

1) Considered that AMRs were placed into an *apriori* unknown environment. The region is assumed to be covered by a WSN to sense for a target. The node closest to the target marks itself as a *target-node* and initiates a packet exchange via a flooding mechanism.
2) Assigned a magnitude (termed *pseu_g*) to each sensor node. That magnitude is a function of the node's communication distance from the *target-node*. So, the *target-node* has the highest magnitude assigned to it. Utilizing network topology and RSS at each sensor node, the pseudo-gradient (P-G) algorithm [4] produces an artificial magnitude distribution in the WSN.
3) Had the AMRs follow the direction of the increasing magnitude to reach the target from any location within this region. The P-G Navigation algorithm [4] determines the direction of AMR navigation.

The technique does not utilize global position information. This allows the P-G algorithm to be used without sophisticated hardware like global positioning systems (GPS). It also means that the algorithm can be used inside buildings, forested areas or inaccessible areas.

In [4], AMRs plan their motion to the *target-node* by traveling node-to-node in straight line paths, using RSS only. Since the motion space is discretized by how the WSN nodes are distributed, the advantage of the method is that the AMR only concerns itself with the node locations as way-points. Having no knowledge of the global optimum path, it ignores any area between these way-points. This also serves as a disadvantage for the method, as this non-represented area between neighboring sensor nodes may further optimize the trajectory of the AMR.

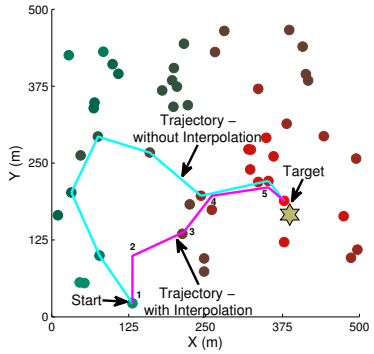
This paper utilizes a *radial basis function* based implicit surface interpolation scheme that improves on the AMR navigation scheme described in [4]. Without the global knowledge of the goal location, the AMR is constrained to move in its local neighborhood. The interpolation scheme approximates the artificial magnitude over this neighborhood, by constructing a surface fit using the *pseu_g* values at neighboring nodes. This allows the AMR to compute a local way-point that utilizes the non-represented space, thereby minimizing the overall trajectory. Figure (1) illustrates the concept showing comparative paths for the both the *interpolated* and *non-interpolated* schemes. As the AMR moves through the numbered locations (instead of nodes) by interpolating the non-represented space, it is able to follow a different trajectory to the *target-node*. The mechanism to reduce the length of this trajectory is discussed in this paper.

The remainder of the paper is organized as follows: Section II discusses related work in this area. Section III discusses the assumptions and system model for the proposed scheme. Section IV introduces the pseudo-gradient interpolation mechanism and discusses the algorithm employed. Section V discusses the implementation of the scheme with experimental results, along with a comparative analysis. Section VI concludes the paper with a description of future work.

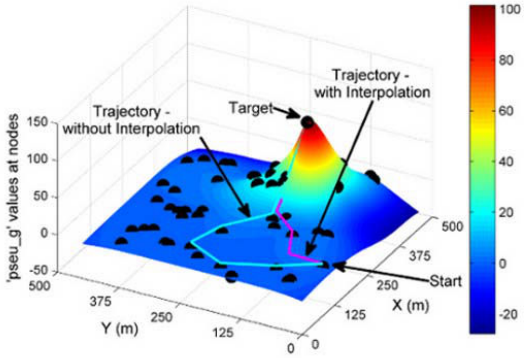
978-1-4673-1405-3/12/$31.00 ©2012 IEEE

II. Related Work

The research in this paper was inspired by the research into WSN-assisted AMR navigation in [3], [4], [6], [8] and [11]. It falls in the position-unaware category; the algorithms are independent of node locations utilizing present network topology, and basing their control strategies on the immediate neighborhood of the nodes only.



(a) Magnitude distribution in the WSN (Numbered locations explained in section (IV-A))



(b) Image depicting interpolated artificial magnitude distribution

Figure 1. AMR Trajectory with and without neighborhood interpolation

Li et al. [8] show the ability of WSNs in acting as guides to navigate AMRs using novel networking protocols and AMR navigation algorithms. The authors developed an artificial potential field based method to navigate AMRs to a goal location by keeping as far away from "dangerous" (obstacle) sites as possible. The scheme utilizes GPS coordinates for the sensor node locations used to assign the artificial potentials. The related research in [1], [6] and [11] utilizes some form of node-to-node navigation, either based on RSS or simple and probabilistic communication hop-count paths. Chen and Henderson [3] can be seen as early proponents of the smart sensor network philosophy using distributed computation and WSN-AMR coordination. Global position awareness is not required as the AMRs utilize the WSN "information field" and the inherent gradient in the source-phenomenon to guide themselves. Although Kotay et al. [7] utilized known global position information, they present algorithms which aggregate global maps of the navigation space, implying that the AMRs do not need to travel node-to-node. Similarly, Severino and Alves [13] demonstrate a centralized scheme where a controller estimates the location of a target using the distance information conveyed to it by the distributed sensor nodes.

Implicit surface reconstruction has been widely used in the field of computer vision [15]. Chaimowicz et al. [2] use implicit surfaces in the field of pattern formation and control of AMR swarms. Implicit surface interpolation in sensor networks for autonomous navigation is a novel approach introduced in this research.

III. System Model

Two critical requirements with respect to WSN-assisted AMR navigation act as performance bounds for the system:

1) There needs to exist a geographic path from a particular starting point for the AMR to traverse and reach the target. The physical region, or without obstacles needs to be such that the target is not completely occluded from the AMR in terms of path traversal.
2) If a geographic path exists, then the WSN deployment shall be such that there exist nodes physically located on or close to that path. The proximity of the nodes to this path, a necessary condition for the algorithm to be useful, depends on how narrow or broad the allowable trajectory for the AMR is.

For the experiments in this paper, the *Log-normal shadowing* model [5] is used to model the relationship between Euclidean distance and RSS accounting for factors such as obstacles, environmental conditions, signal interference, etc. Several studies have extensively characterized the RSS-distance relationship [9], [16]. This paper includes experiments with noise added to the RSS estimates as well as a practical evaluation of the RSS-distance relationship.

IV. Implicit surface based interpolation mechanism

As stated in [15], implicit surfaces can basically be created by summing a set of *radial basis functions* (RBFs). This involves the solution of a linear system, which assigns weights to the individual basis functions. In relation to this research, the *constraints* which are interpolated by the surface are the approximate locations of the sensor nodes in the neighborhood of the AMR based on the RSS values.

The interpolation problem can be stated as:

Given a set of N different points $\{\overrightarrow{\mathrm{x}_i} \in \mathbb{R}^{\mathrm{m}} \mid i = 1, 2, ..., N\}$ and a corresponding set of N real numbers $\{d_i \in \mathbb{R}^1 \mid i = 1, 2, ..., N\}$, find a function $F : \mathbb{R}^m \rightarrow \mathbb{R}^1$ that satisfies the interpolation condition:

$$F(\overrightarrow{\mathrm{x}_i}) = d_i, \qquad i = 1, 2, ..., N \tag{1}$$

Equation (1) involves choosing a function F given by [15]:

$$F(\overrightarrow{\mathrm{x}}) = \sum_{i=1}^{N} w_i \, \varphi(\|\overrightarrow{\mathrm{x}} - \overrightarrow{\mathrm{x}_i}\|) \tag{2}$$

where $\overrightarrow{x_i}$ are known data points (i.e., estimated locations of the neighboring sensor nodes), and $w_i \in \mathbb{R}^1$ are the weights assigned to individual RBFs. Combining the equations gives a linear system:

$$\mathbf{\Phi}\overrightarrow{w} = \overrightarrow{d} \tag{3}$$

$\mathbf{\Phi}$ is called the *interpolation* matrix, $\overrightarrow{w}$ is called the *linear weight* vector and $\overrightarrow{d}$ is called the *desired response* vector. With respect to this research, $\overrightarrow{d}$ is the vector of magnitudes *pseu_g*, at each of the neighboring sensor nodes. Equation (3) is a linear system of equations in the unknown $\overrightarrow{w}$. In order to determine $\overrightarrow{w}$, the *method of conjugate gradients* (C-G) is adopted. [14] presents a straightforward, iterative algorithm for solving linear systems using the method of *conjugate-gradients*.

A. Iterative Implicit Surface Interpolation

It is treated as a two-phase supervised learning problem:

- The training phase involves the solution of the linear system to obtain $\overrightarrow{w}$.
- The generalization phase interpolates between the known data points, along the fitted surface using $\overrightarrow{w}$.

The *training* phase consists of estimating the *linear weight* vector $\overrightarrow{w}$ from: (i) a known $\overrightarrow{d}$ and (ii) an *interpolation* matrix $\mathbf{\Phi}$, constructed using known data-points $\overrightarrow{x_i}$. This is described in the next section.

1) At each location, using its directional antennas and the RSS values, the AMR estimates the angle and distance of the *N* nodes in its neighborhood[1].
2) The AMR then assigns a local coordinate frame to its neighborhood. The AMR's location becomes the origin (0,0), and based on the distance and angle information, appropriate (x, y) coordinates are assigned to the neighboring sensor nodes. Each value d_i in $\overrightarrow{d}$, is the pseudo-gradient magnitude $pseu\_g$ for each neighboring node *i* of *N*.
3) *Choice of RBF*: $\mathbf{\Phi}$ is constructed using $\varphi(.)$ (Table (I)). The symmetry of $\mathbf{\Phi}$ results in only $^{N(N-1)}/_{2}$ entries to be calculated, instead of N^2.

Table I
RADIAL BASIS FUNCTION EXAMPLES

Function Name	$\varphi(r) \ldots r \in \mathbb{R}$
Gaussian	$\exp(-\frac{r^2}{2\sigma^2}) \ldots \sigma > 0$
Inverse Multiquadric	$\frac{1}{(r^2+\sigma^2)^{1/2}} \ldots \sigma > 0$
Thin Plate Spline (TPS)	$r^2 \cdot \log r$

The *generalization* phase uses $\overrightarrow{w}$ from the *training* phase to determine the *desired response* vector.

[1]The neighborhood is based on the wireless communication range of the AMR.

1) The neighborhood of the AMR is represented as a $\sqrt{n}$ x $\sqrt{n}$ grid of n points.
2) Using the weights w_i, the d_j values are calculated for each $\overrightarrow{x_j}$ using equation (3), $\{\forall\, i\ neighbors;\ j = 1 \rightarrow n\}$.

This newly formed $\overrightarrow{d}$ has a *pseu_g*-value assigned to each of the *n* points in the AMR's neighborhood. The AMR then determines the point which has the highest *pseu_g*-value and moves to it. Figure (1)(b) is an example of the interpolated surface using *Thin Plate Spline* (TPS) RBF. At each of the intermediate locations (numbered 1 through 5 in Figure (1)(a)), the AMR executes the *training* and *generalization* phases in order to determine the next way-point in the trajectory.

B. Analysis of Iterative Implicit Surface Interpolation

For analysis, the *training* phase was run on a set of data-points generated uniform randomly over a region, one having an area equal to that covered by the communication range of an off-the-shelf available wireless sensor. Each data-point was assigned a *pseu_g*-value, uniform randomly generated over the interval [0, 1], forming $\overrightarrow{d}$. From [14], the important settings for the training are the values of:

1) Epsilon ε for the precision of the surface fit. $\varepsilon = 1 \times 10^{-10}$.
2) Number of iterations i_{max}, for how long the algorithm runs before the training fails. $i_{max} = 5000$.

The *training* was run for an increasing number of data-points in the region, with 30 different random generations for each training set. Figure (2) shows the variation in the performance of the *training* phase. As the number of data-points in the neighborhood increases, the number of iterations required for $\overrightarrow{w}$ to be trained to ε-precision, also increases.

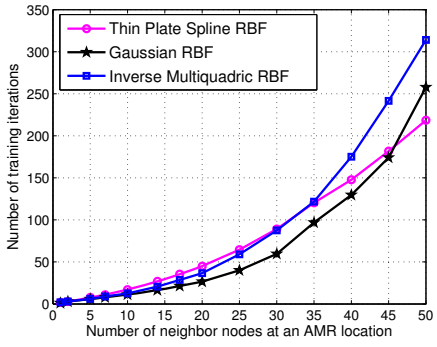
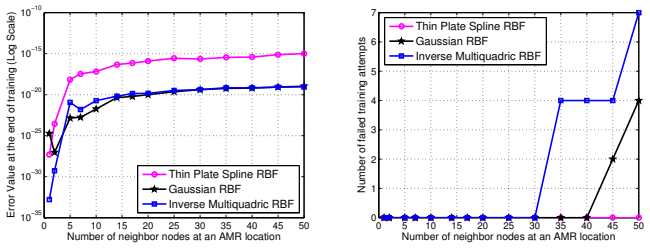


Figure 2. Performance of the Conjugate Gradients Algorithm Training

Figure (3)(a) shows the variation of the δ-value with increasing number of data-points. The error follows an expected trend. Although it increases rapidly initially, and then at a slower rate, it also takes more iterations to train. Figure (3)(b) shows that the number of training attempts that fail to achieve the set precision within the i_{max} iterations, also increases with increasing number of data-points. It is observed from both Figures (2) and (3) that, an increase in the number of training

points deteriorates the performance. This is attributed to the fact that the generated implicit surface, being constrained to pass through each of the *training* data points, becomes suboptimal as it attempts to fit an increasing number of them, thereby degrading the output of the training phase.



(a) Variation in δ **(b)** Failed Training Attempts

Figure 3. Performance analysis of the training phase

V. Experiments

To analyze the effectiveness of the algorithm, two key parameters are: (i) travel-distance for the AMR to reach a target, and (ii) the number of nodes required in the AMR trajectory from the start location to the target. The travel-distance parameter is measured as the ratio of the actual distance traveled by the AMR to the Euclidean distance between the start location and the target.

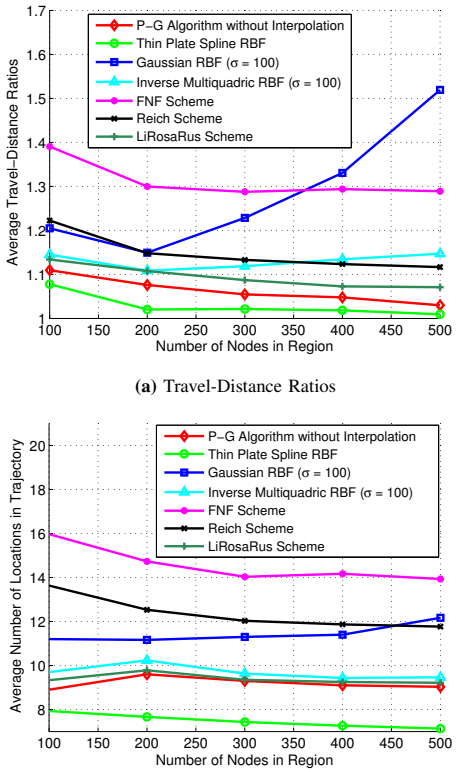
A. Navigation Effectiveness

Figure (4) shows the performance of the three different RBFs with respect to the two parameters[2]. The simulations were conducted with 30 different random generations of node locations. As seen, the performance of the introduced mechanism compares favorably with existing mechanisms in [6] (FNF Scheme), [8] (LiRosaRus Scheme), and [11] (Reich Scheme). Although, the Inverse Multiquadric RBF and the Gaussian RBF perform worse than the original P-G algorithm, subsequent testing showed their trajectories to be sensitive to the 'σ' parameter (Table (I)) Higher values of 'σ' produces trajectories similar to the ones without interpolation (Figure (5)). The Thin Plate Spline RBF though, offers a significant improvement over all the other techniques.

A few important points are noted from Figure (4). The density of the WSN impacts the performance of the interpolation scheme. The data generally follows an inverted bell curve for the RBFs.

1) With a low number of nodes in the region, there is a very low number of neighborhood training points for the *training* phase, which in turn generates a suboptimal interpolated surface fit for the neighborhood.
2) As the number of nodes in the region increases, the number of training points in the neighborhood also increases, improving the interpolated gradient surface, thereby reducing the navigation distance.

[2]For the interpolated navigation, the 'Number of Nodes' translates to number of way-points in trajectory.



(a) Travel-Distance Ratios

(b) Number of Nodes in Trajectory

Figure 4. Performance Analysis

3) Beyond 300 nodes, the improvement in performance reduces dramatically. This result is consistent with that observed in Figures (2) and (3). Here, with increasing number of WSN nodes, the deployed region is also better represented by the node locations. This reduces the advantage offered by interpolation, generating similar way-points as with no interpolation.

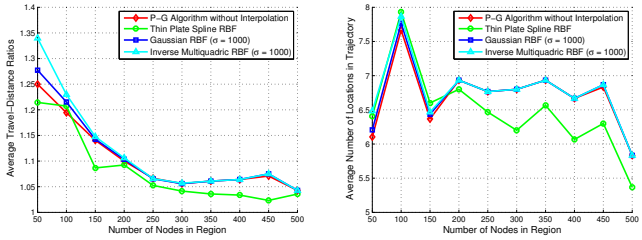
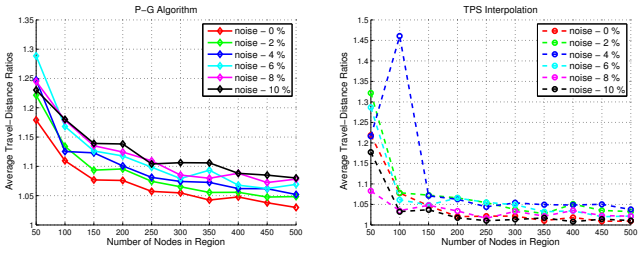


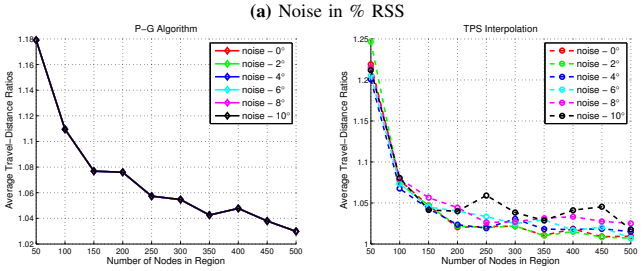
Figure 5. Trajectory Analysis for Interpolation schemes with $\sigma = 1000$

B. Performance with Noise Added

Although ideal in simulation, the estimation of neighbor-node angles and distance is prone to noise in the real world. The performance of the TPS RBF is compared to the original P-G Algorithm, in the presence of noise in RSS as well as angle estimations. Noise is introduced as a normally distributed function with standard deviation σ. The σ for angle is $0^o - 10^o$ in steps of 2^o. The σ for RSS is 0-10% in steps of 2%.



(a) Noise in % RSS



(b) Noise in degrees of Angle

Figure 6. Performance Analysis of TPS RBF with Noise added

Figure (6) shows the performance over 30 random generations for node locations. It is observed from the data that the TPS RBF still performs better than the original P-G algorithm without interpolation. It is expected that as the noise levels increase, the performance gets worse. The angle noise does not affect the P-G algorithm since it does not use bearing information for navigation. This is evident from both the plots in the figure. There is an inconsistency observed for the 10% RSS Noise data in TPS RBF. This is attributed to the failed training and navigation attempts which are not accounted for in calculating the trajectory parameters. The data for the failed attempts is not included in this paper.

C. Hardware Experiments

The particular directional antenna used for experimentation in this research is shown in Figure (7)(a) [10]. Based on its radiation pattern, a 120^o offset positioning of three directional antennas maximizes the coverage around the AMR, as shown in Figure (7)(b).

(a) Directional Antenna (b) AMR with directional antennas

Figure 7. AMR Hardware

The variation in RSS at the directional antennas was tested using a stationary wireless transmitter node at 3 distances - 5 feet, 10 feet, and 15 feet. For each distance, the AMR was rotated counter-clockwise[3], in place, in steps of 30^o, for the full 360^o. Figure (8)(a) shows a sample of the tests - the RSS values at the three antennas for a distance of 15 feet. It was observed that the readings followed the expected trend for counter-clockwise rotation. Additionally, as the distance increased, the absolute RSS values reduced.

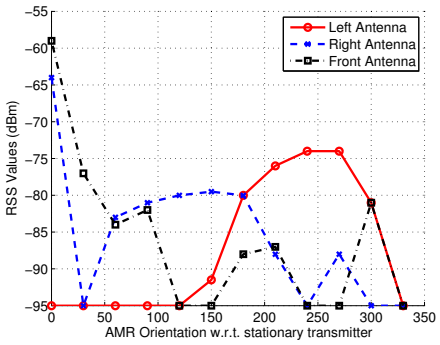
A weighted triangulation mechanism is used to estimate the bearing of the incoming wireless signal. Based on the RSS values at each directional antenna, two antennas with the higher values are chosen. Algorithm (1) shows the bearing estimation example for the 'Front' and 'Left' antennas being the antennas with the higher RSS values. The algorithm biases the bearing towards one antenna more than the other, based on the actual RSS values at the antennas. The *turnThreshold* parameter, used to determine this biasing, is based on the maximum possible difference in the RSS values at the directional antennas. It is based on the directional antenna gains - a value of -12 dB is used here. This value is generally determined through experimentation. Figure (8)(b) depicts the bearing estimation using the algorithm averaged over 3 experiments.

Algorithm 1 RSS Angle Estimation (in degrees)

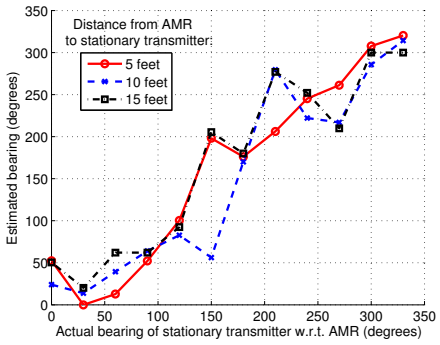
1: biasR = 0.5 + $\left(0.5 \cdot \left[\frac{RSS_{Left} - RSS_{Front}}{turnThreshold}\right]\right)$
2: biasL = 1 - biasR
3: biasFactor = $\left[\frac{(\text{biasR} \cdot RSS_{Left} - \text{biasL} \cdot RSS_{Front})}{(\text{biasR} \cdot RSS_{Left} + \text{biasL} \cdot RSS_{Front})}\right]$
4: Bearing = biasFactor·60 + 60

To investigate the performance of the scheme in hardware, preliminary experiments consisted of three tests - one indoors and two outdoors. Eight wireless nodes were placed equidistant from the AMR in a circle - separated by 45^o. These nodes act as neighbors for the AMR in its current location. Each sensor node was assigned a *pseu_g*-value, uniform randomly generated over the interval [0, 1], forming $\overrightarrow{d}$. This $\overrightarrow{d}$ was kept common across the three experiments. The AMR was then commanded to estimate the range and bearing to each of the nodes and use the interpolation scheme to calculate the next way-point for this neighborhood. In each test, RSS values from all the neighbors, averaged over 3 seconds, were noted at the AMR, for one omnidirectional antenna and the three directional antennas. To evaluate the performance of the interpolation scheme, TPS RBF was used. The AMR-to-node physical distance was: 6 ft. (Indoor); 7.5 ft. (Outdoor-1); and 15 ft. (Outdoor-2). Table (II) shows a comparison of the way-point (range and bearing from the current AMR position) generated. For each test, the way-point was generated using three possible methods: M1 - simulation, using ideal physical distances and angles; M2 - using RSS-distance conversion with Log-normal shadowing, but using ideal angle information; and M3 - using RSS-distance conversion with Log-normal shadowing, and using the angle information from Algorithm (1).

[3] 0^o is with the AMR's front antenna pointing in the direction of the stationary transmitter.
The TMote Sky nodes were used in the experiments [12].



(a) RSS Values at 15 Feet



(b) Bearing estimate using Algorithm (1)

Figure 8. Directional Antenna Experiments

Table II
DATA FROM HARDWARE EXPERIMENTS

	Way-point (range [ft.], bearing [degrees])		
Method	Indoor	Outdoor-1	Outdoor-2
M1	8.44, 108.44	7.50, 90.00	15.03, 93.81
M2	9.71, 101.56	10.05, 98.12	15.89, 90.44
M3	11.71, 115.99	11.91, 79.27	16.55, 74.03

Considering the way-point from (M1) as the basis for comparison, as expected, method (M3) shows the worst performance. The RSS-based distance estimates combined with the angle estimation using the directional antennas, suffer from environmental variations impacting the accuracy. A particle filter based approach is proposed in the future extension of this research, similar to [9]. In this, the AMR probabilistically estimates a way-point from multiple possible way-points, and improves the estimate by updating the range and bearing information in real-time as it moves in its neighborhood.

VI. Conclusions and Future Work

In this paper, an improvement over the scheme presented in [4] is demonstrated, using interpolated implicit surfaces in the AMR neighborhood. Simulation experiments show that the scheme significantly reduces the navigation distance to the target. The performance of the scheme is analyzed in terms of the computational requirements as well as under noisy conditions. Preliminary hardware experiments bear out the result.

The advantage of the interpolation scheme is more significant in the obstacle avoidance scenario. Attenuation of the RSS is a strong indication of the presence of an obstacle in the communication path, which alters the *pseu_g*-values. The AMR can navigate around the obstacle by generating intermediate way-points in the neighborhood using an altered implicit surface. This phenomenon shall be examined further. Some of the other key aspects of the algorithm that will be analyzed in future research are: (i) further tests with WSN and AMR, and (i) robustness to link and node failures. The research shall include analysis for multiple target presence and coordinated AMR navigation as well.

References

[1] M. A. Batalin, G. S. Sukhatme, and M. Hattig, "Mobile Robot Navigation using a Sensor Network," in *Proc. IEEE Intl. Conf. on Robotics and Automation, (ICRA '04)*, 2004, pp. 636–641.

[2] L. Chaimowicz, N. Michael, and V. Kumar, "Controlling Swarms of Robots using Interpolated Implicit Functions," in *Proc. IEEE Intl. Conf. on Robotics and Automation, (ICRA '05)*, 2005, pp. 2487–2492.

[3] Y. Chen and T. C. Henderson, "S-NETS: Smart Sensor Networks," in *Proc. Intl. Symp. on Experimental Robotics, (ISER '00)*. Hawaii: Springer-Verlag, Dec. 2000, pp. 81–90.

[4] N. Deshpande, E. Grant, and T. C. Henderson, "Experiments with a "pseudo-gradient" algorithm for target localization using wireless sensor networks," in *Proc. IEEE Intl. Conf. on Multisensor Fusion and Integration for Intelligent Systems, (MFI 2010)*, Sept. 2010, pp. 74–79.

[5] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.

[6] J.-R. Jiang, Y.-L. Lai, and F.-C. Deng, "Mobile Robot Coordination and Navigation with Directional Antennas in Positionless wireless sensor networks," in *Proc. Intl. Conf. on Mobile Technology, Applications, and Systems, (Mobility '08)*. ACM Press, 2008, pp. 1–7.

[7] K. Kotay, R. Peterson, and D. Rus, "Experiments with robots and sensor networks for Mapping and Navigation," in *Proc. Intl. Conf. on Field and Service Robotics*, Australia, 2005.

[8] Q. Li, M. De Rosa, and D. Rus, "Distributed Algorithms for Guiding Navigation across a sensor network," in *Proc. 9^{th} Intl. Conf. on Mobile Computing and Networking, (MobiCom '03)*. ACM Press, 2003, pp. 313–325.

[9] D. Lymberopoulos, Q. Lindsey, and A. Savvides, "An Empirical Characterization of Radio Signal Strength Variability in 3-D IEEE 802.15.4 Networks Using Monopole Antennas," in *Proc. 3^{rd} European Workshop, (EWSN 2006)*, ser. LNCS, H. K. K. Romer and F. Mattern, Eds. Zurich, Switzerland: Springer-Verlag, Feb. 2006, vol. 3868, pp. 326–341.

[10] Quatech. (2003) ACH2-AT-DP006 Directional Antenna. DPAC Technologies. [Online]. Available: http://www.quatech.com/catalog/accessories.php.

[11] J. Reich and E. Sklar, "Robot-Sensor Networks for Search and Rescue," in *Proc. IEEE Intl. Workshop on Safety, Security and Rescue Robotics*, 2006.

[12] Sentilla. (2007) Moteiv Tmote Sky Datasheet. Sentilla Corporation. [Online]. Available: http://www.sentilla.com/moteiv-transition.html.

[13] R. Severino and M. Alves, "Engineering a Search and Rescue Application with a wireless sensor network - based Localization Mechanism," in *Proc. IEEE Intl. Symp. on a World of Wireless, Mobile and Multimedia Networks, (WoWMoM 2007)*, 2007, pp. 1–4.

[14] J. R. Shewchuk, "An Introduction to the Conjugate Gradient Method Without the Agonizing Pain," School of Computer Science, Carnegie Mellon University, Tech. Rep., 1994.

[15] G. Turk, H. Q. Dinh, J. F. O'Brien, and G. Yngve, "Implicit Surfaces that Interpolate," in *Proc. Intl. Conf. on Shape Modeling and Applications, (SMI 2001)*, 2001, pp. 62–71.

[16] S. Yang and H. Cha, "An Empirical Study of Antenna Characteristics Toward RF-Based Localization for IEEE 802.15.4 Sensor Nodes," in *Proc. 4^{th} European Workshop, (EWSN 2007)*, ser. LNCS, K. Langendoen and T. Voigt, Eds. Springer-Verlag, 2007, vol. 4373, pp. 309–324.